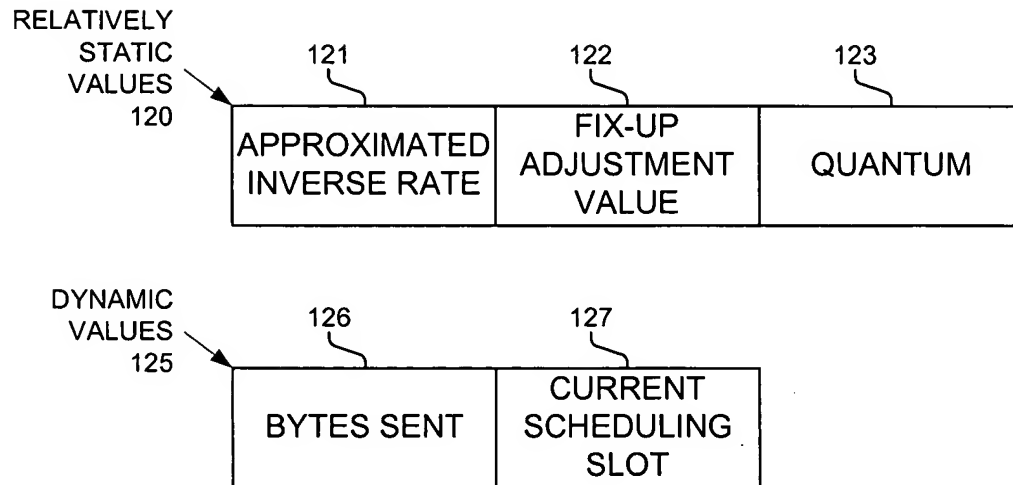


FIGURE 1A

+



STORED VALUES
FIGURE 1B

+

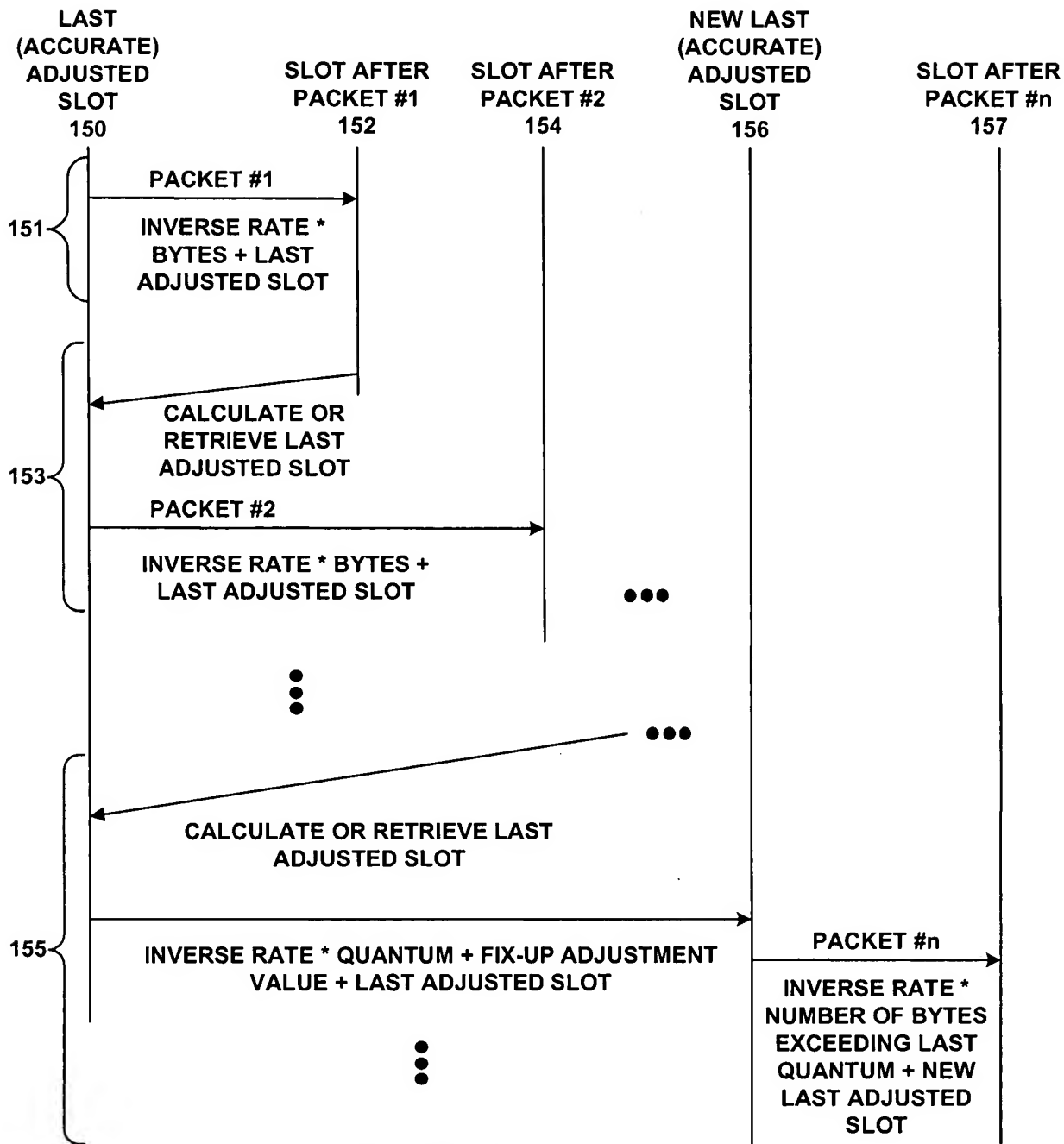
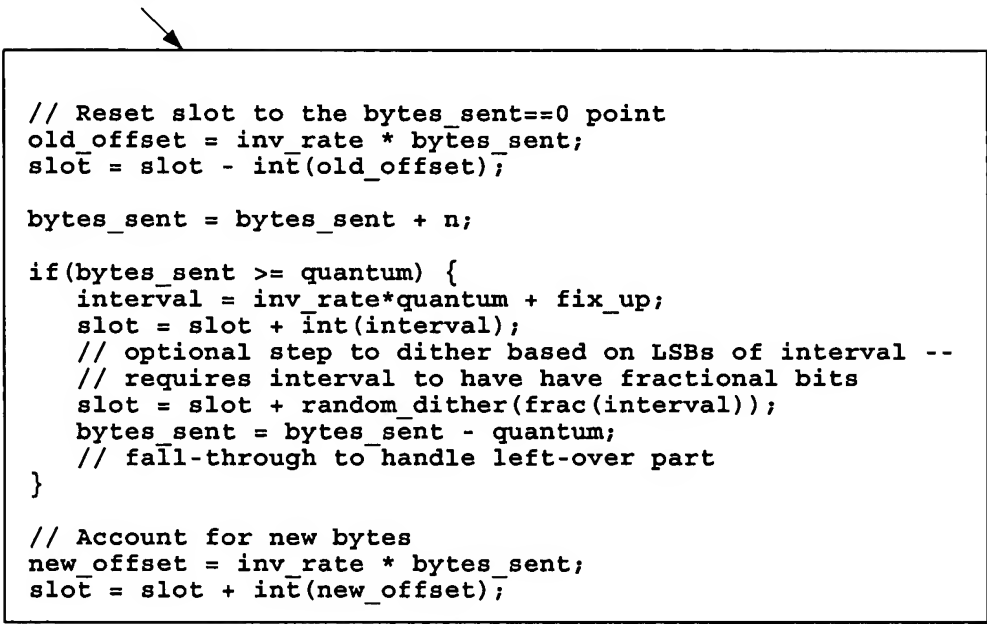


FIGURE 1C

PSEUDO-CODE

180



```
// Reset slot to the bytes_sent==0 point
old_offset = inv_rate * bytes_sent;
slot = slot - int(old_offset);

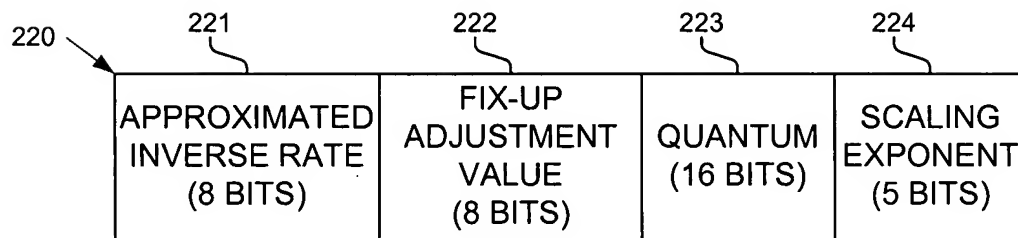
bytes_sent = bytes_sent + n;

if(bytes_sent >= quantum) {
    interval = inv_rate*quantum + fix_up;
    slot = slot + int(interval);
    // optional step to dither based on LSBs of interval --
    // requires interval to have have fractional bits
    slot = slot + random_dither(frac(interval));
    bytes_sent = bytes_sent - quantum;
    // fall-through to handle left-over part
}

// Account for new bytes
new_offset = inv_rate * bytes_sent;
slot = slot + int(new_offset);
```

FIGURE 1D

+



**RELATIVELY STATIC STORED
VALUES FOR RATE COMPUTATION
FIGURE 2**

+

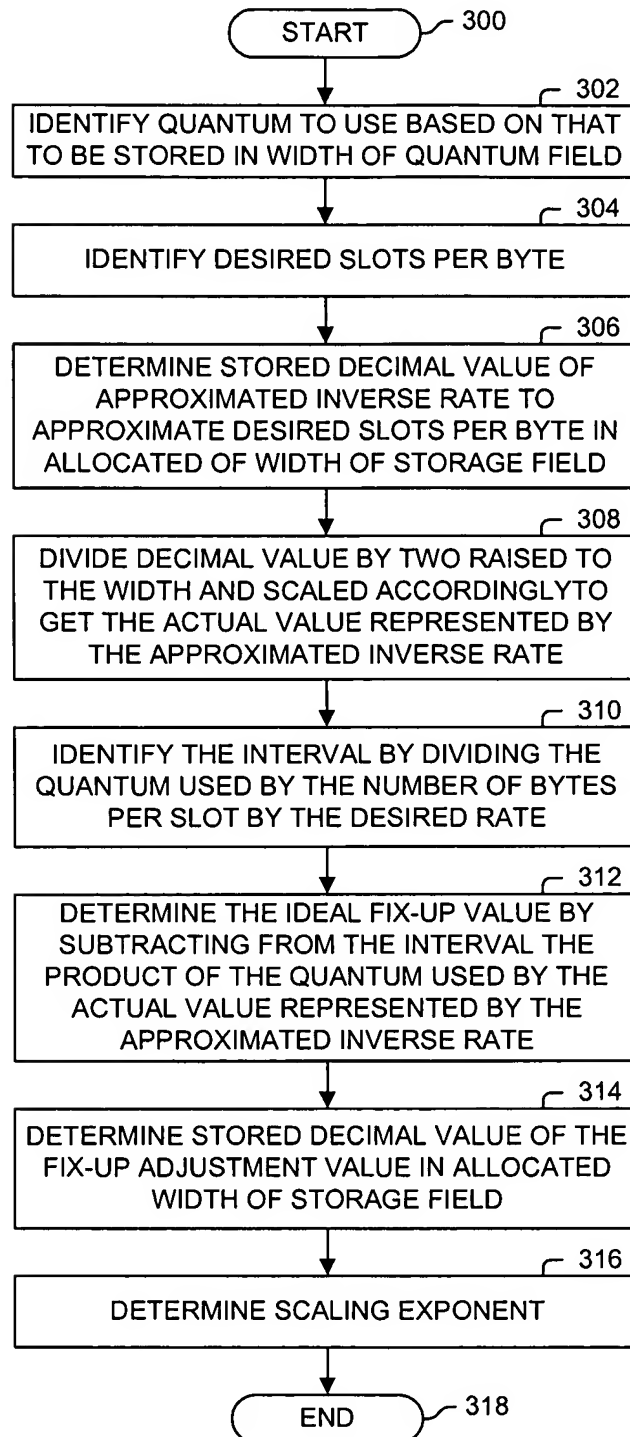


FIGURE 3A

+

+

TARGET RATE (%)	ACTUAL RATE (%)	INTERVAL	QUANTUM BYTES	SLOTS / BYTE	APPROX. INVERSE RATE	FIX-UP ADJUSTMENT VALUE
640	640.00689	1.020000	65280	0.0000156	$131/256 * 2^{-15}$	$18/256 * 2^{-7}$
300	300.00330	1.092267	32768	0.0000333	$139/256 * 2^{-14}$	$207/256 * 2^{-7}$
200	200.00034	1.638400	32768	0.0000500	$209/256 * 2^{-14}$	$183/256 * 2^{-7}$
100	100.00017	1.638400	16384	0.0001000	$209/256 * 2^{-13}$	$183/256 * 2^{-7}$
90	90.00005	1.820444	16384	0.0001111	$233/256 * 2^{-13}$	$4/256 * 2^{-7}$
75	75.00083	2.184533	16384	0.0001333	$139/256 * 2^{-12}$	$207/256 * 2^{-6}$
50	50.00009	3.276800	16384	0.0002000	$209/256 * 2^{-12}$	$183/256 * 2^{-6}$
25	25.00004	6.553600	16384	0.0004000	$209/256 * 2^{-11}$	$183/256 * 2^{-5}$
10	10.00013	16.384000	16384	0.0010000	$131/256 * 2^{-9}$	$18/256 * 2^{-3}$
1	1.00000	163.840000	16384	0.0100000	$163/256 * 2^{-6}$	$215/256 * 2^0$
0.1	0.10000	1638.400000	16384	0.1000000	$204/256 * 2^{-3}$	$205/256 * 2^{-3}$
0.012207	0.0122069	8192.000000	10000	0.8192000	$209/256 * 2^0$	$112/256 * 2^6$

**EXAMPLES OF RATE ENCODING
(10KB MTU, 8192 SLOTS)
FIGURE 3B**

+

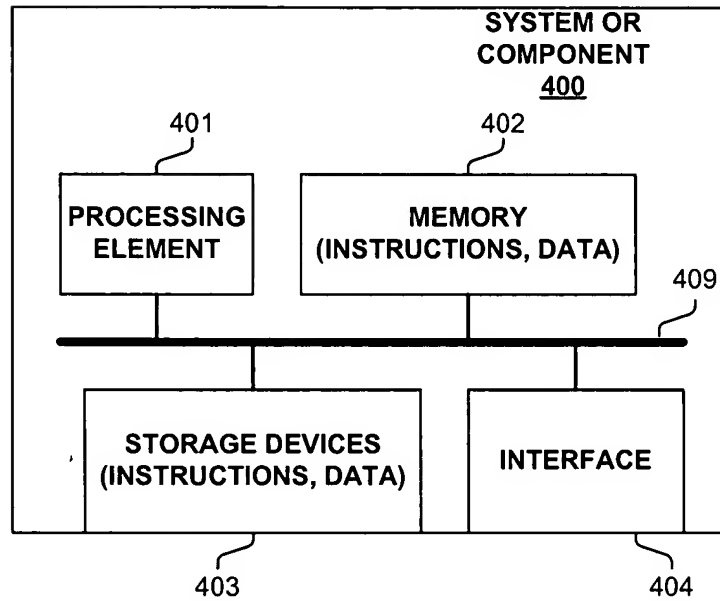


FIGURE 4A

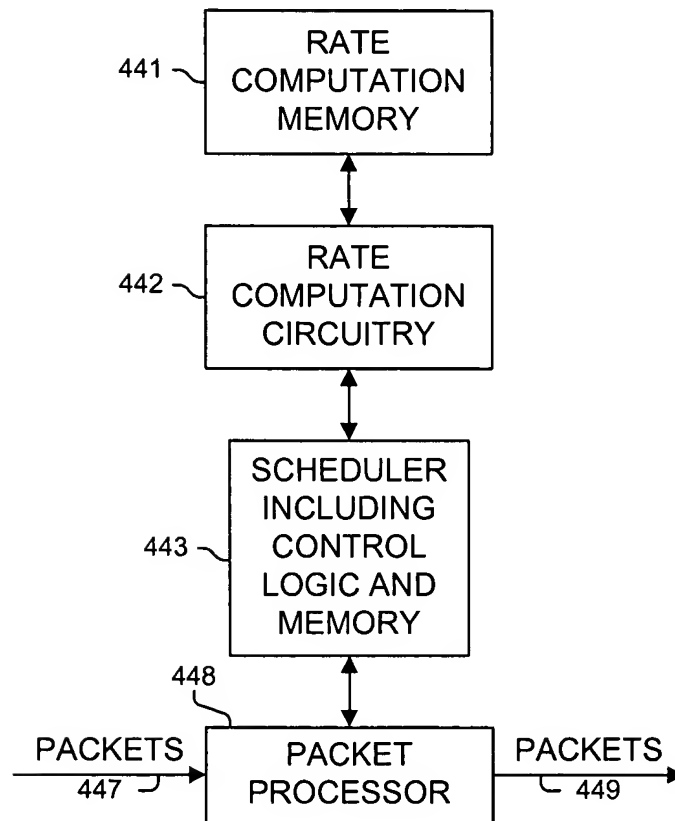


FIGURE 4B